

**METHOD AND SYSTEM FOR GENERATING MATERIALS FOR
PRESENTATION ON A NON-FRAME CAPABLE WEB BROWSER**

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a system and method for presenting, over a network, materials contained in a single computer file.

2. Description of the Related Art

The Internet, as it is popularly known, has become an important and useful tool for accessing a wide variety of information. One component of the Internet is the World Wide Web (hereinafter, the web). In recent years the web has become an increasingly popular vehicle for providing information to virtually anyone with access to the Internet. Many sites (hereinafter, web sites) have been established to provide information in many different forms, such as text, graphics, video and audio formats over the web.

A typical web site includes system and application software programs installed on a web server that is connected to the Internet. By connecting the web server to the Internet, clients that are connected to the Internet can access the web site via the web server. Usually a client is located remotely from the web server,

1 although the client and server can be at the same location. Also, a web server can
2 be connected to a private intranet, as opposed to or in addition to the public
3 Internet, in order to make a web site privately available to clients within an
4 organization.

5 A client typically accesses the web site by using a web browser. The web
6 browser is a software program which runs on the client and receives from the
7 server information formatted in a known manner. A very popular format for
8 information sent over the web from a server to a client, is the hypertext mark-up
9 language (HTML). HTML is a tag based script format, in which tags surround the
10 information to be presented. By tagging the information to be sent to a browser,
11 the browser can interpret the tags and handle the presentation of the information
12 sent from the server. It is left to the web browser, at the client, to determine the
13 specific formatting of the information, based on the tags included in the HTML
14 information sent from the server. For example, information to be displayed at a
15 client might include header information followed by a list of other information.
16 Fig. 1A shows a portion of HTML code with such a header and list. When the
17 HTML information is received by a client and interpreted by a web browser, the
18 information is displayed, as is shown in Fig. 1B, for example, in which a header 10
19 and list 12 are presented to a user at the client.

20 The HTML information sent by the server does not specify the particular
21 size, font and placement of the header 10 and the list 12. Rather, the header and

list information are surrounded by HTML tags which identify that information as a header and a list, respectively. It is the responsibility of the web browser at the client to determine how the information is presented.

Over the past several years, HTML-compliant web browsers have proliferated at a great rate. Accordingly, a large install base of HTML-compliant web browsers has been established worldwide.

HTML supports the use of links, which allow a web browser user to link from one source of information to another, easily and rapidly. A link provides a user with the means to navigate the web, and even navigate within a specific web site. The server, when sending HTML information to a web browser, would include a Uniform Resource Locator (URL) in information associated with the link. Typically, a link would be displayed to a user in a special format, such as being highlighted. The user could select that link, for example, by moving a pointing device such as a computer mouse to position a cursor over the link and selecting that link by pressing a button on the mouse. When a user selects a link, the web browser sends a request for the corresponding URL. The URL identifies a specific web resource, such as a specific web page at the web site. The server receives the request for the URL, determines the information to be sent to the client, accesses that information and then sends that new information to the client for presentation. The client receives the new presentation information sent by the server, which is typically HTML formatted information, and presents it to the user

NAVIGATOR 3.0, allows a client to display, via the web browser, more than one frame at a time. This feature allows a user to display one set of information in one frame, while displaying another set of information in another frame, so that both frames are displayed by the web browser at the same time. Fig. 2 shows an example of a display generated by a frame-capable web browser. The display shown in Fig. 2 includes four frames of information displayed at the same time. As depicted in Fig. 2, a table of contents frame 20, shown on the left-hand side of the display, would list items related to the table of contents of a document. A header frame 22, shown in the upper portion the display, would contain information identifying the information shown in a main frame 24, depicted in the middle of the display of Fig. 2. The main frame 24 would contain detailed information related to the item selected in the table of contents frame 20. A footer frame 26 may also be included to present information at the bottom of the display, such as footnotes, or a number of links, for example.

In this frame-based environment if a user selects one of the items contained in the table of contents frame, a request for a URL is sent to the web server to direct it to send new information to be displayed in the main frame relating to that specific URL. For example, if the user selects an item in the table of contents frame, the web browser, in response, sends to the web server a request for a URL unique to that selected item. The web server receives the request for the URL which might identify a separate computer file on the server. The information in

that file is transmitted to the web browser and the web browser displays the received information in the main frame.

In order to manage and identify the frames in which the information is to be displayed, the server requires that specific frames be identified in the HTML structure of the files at the server. For example, Table 1 shows an example of HTML source code necessary for creating the frames shown in Fig. 2. For example, frame tags are required in the HTML source code to identify a frame. Those frame tags include a frame source name (SRC) and attributes about the frame such as the location for display by the web browser (e.g., row and column sizes). Here for example, the table of contents frame is located in a separate file named TOC.HTM. The frame tag identifies the size of the frame and location of the frame. Similarly, the frame tags for the header, main and footer frames identify those frames as well as the relative positions of those frames for display by the web browser. Upon receiving these special frame tags, the frame-capable web browser decodes those frame tags. The web browser then displays the information relating to each frame in the corresponding frame. That information is displayed in areas at the location and relative size specified by the frame tags.

1 Although use of frames provides another dimension of functionality for
2 display of web-based information, web sites that use frames require web browsers
3 to support the frame feature, in order to have information presented in the frames.
4 While displaying information according to a frame paradigm can be useful in
5 displaying web-based information, the size of the install base of non-frame-capable
6 web browsers limits the use of frames. That is, in order to use frames, a frame-
7 capable web browser must be installed on the client. However, because of the
8 large number of installed web browsers which are not frame-capable, use of frames
9 on web sites is often avoided, or else HTML code is included to generate both
10 frame-based and non-frame-based web pages for transmission to frame capable and
11 non-frame-capable web browsers. Creating two sets of data, one for frame-
12 capable browsers and another for non-frame capable browsers, creates difficulties
13 in managing and maintaining the redundant information.

14 Another problem with the use of presenting information in frames, is that a
15 single, unified background image cannot be presented by a web-browser that
16 presents frames of information.

17 SUMMARY OF THE INVENTION

18 The present invention is directed to solving the above problems. That is,
19 the present invention is directed to presenting information over the World Wide
20 Web in a manner such that information from different files or from the same file

1 information is maintained, yet that information is used in generating a plurality of
2 the pages of presentation material.

3 BRIEF DESCRIPTION OF THE DRAWINGS

4 The above objects and advantages of the present invention will become
5 more apparent by describing in detail a preferred embodiment thereof with
6 reference to the attached drawings in which:

7 **Figs. 1A and B** respectively depict HTML source code for a web page and
8 a display generated by a web browser based on that source code, as is known in
9 the prior art;

10 **Fig. 2** depicts frames produced by a frame-capable web browser, as is
11 known in the prior art;

12 **Fig. 3** is a block diagram of a client/server system for transmitting
13 presentation materials from a web-based server to a client according to the
14 invention;

15 **Fig. 4** shows presentation of materials on a web browser according to the
16 invention;

17 **Fig. 5** is a flowchart for describing a process of generating the presentation
18 materials shown in Fig. 4;

19 **Fig. 6** shows presentation of other materials on a web browser according
20 to the invention;

1 **Fig. 7** shows a display of materials with pseudo-frames, according to the
2 invention; and

3 **Fig. 8** shows a plurality of presentation areas with a uniform background
4 across all the presentation areas.

5 **DETAILED DESCRIPTION OF THE INVENTION**

6 A preferred embodiment of a method, system and article of manufacture
7 for managing internet presentation materials in a single file format according to the
8 present invention is described below in detail with reference to the accompanying
9 drawings.

10 The present invention is directed to a method, system and article of
11 manufacture for managing internet presentation materials in a single file format.
12 The invention takes advantage of properties of a dynamic HTTP application
13 software 32 program which runs on a web server 30, as shown in Fig. 3. An
14 example of a preferred dynamic HTTP application is NET.DATA manufactured by
15 IBM Corp., although, other programs capable of dynamically creating
16 internet/intranet information can be used. The dynamic HTTP application 32
17 allows information on a web page to be created or changed dynamically, while the
18 application program is running, as opposed to maintaining merely static
19 information which can be changed only by a system administrator, or similarly

Table 2: Web Macro with a Plurality of HTML Sections

```
%define{
  header = {
    <center>IBM Internet Yellow Pages</center>
  }
  contents = {
    <b>Table of Contents</b>
    <ul>
      <li><a href="#">Purpose</a>
      <li><a href="#">Overview</a>
      <li>Hot <a href="#">New Technology</a>
      •
      •
      •
    </ul>
  }
  purpose = {
    <b><i>Purpose</i></b>
    <blockquote>
      To rapidly communicate material and ...
      <p>This internet site is updated ...
    </blockquote>
  }
  overview = {
    <b><i>Overview</i></b>
    <blockquote>
      IBM's Internet Yellow Page Solution is a collection of IBM
      software and hardware products....
      <p>Using IBM's award-winning database, DB2, ...
    </blockquote>
  }
}
```

•

•

•

Table 2 (continued)

```
footer = {  
  <a href="...">IBM home page</a> |  
  <a href="...">Order</a> |  
  <a href="...">Search</a> |
```

```
</BODY></HTML>
```

```
%}
```

```
%}
```

```
%HTML(purpose){
```

```
$(header)
```

```
$(contents)
```

```
$(purpose)
```

```
$(footer)
```

```
%}
```

```
%HTML(overview){
```

```
$(header)
```

```
$(contents)
```

```
$(overview)
```

```
$(footer)
```

```
%}
```

```
%}
```

Variables are defined within the %define section of the web macro. For example, five variables are defined in the %define section of the web macro shown in Table 2. The first variable defined in this example is the variable "header," followed by the variable "contents." Here, the value of the "contents" variable is text describing a table of contents for a series of charts used in a presentation. This example uses charts as the content of the web pages, merely to illustrate that the value of a variable can be essentially any type of information. Here, the Table of Contents includes at least three items, namely, Purpose, Overview and Hot New Technology. The third variable is named "purpose." The value of the variable "purpose" is text describing the purpose of the subject matter of the charts. As shown in Table 2 the value of the variable "purpose" is simply some paragraphs of text information. The fourth variable shown in this web macro is named "overview." Here too, the value of the variable "overview" is text information. The text in the "overview" variable is to be presented on the web page as part of the charts for presentation at the client. The fifth variable in the web macro is named "footer" and includes links for a menu bar displayed at the bottom of the page.

The second section of the web macro is the portion of the macro for generating HTML. In the example shown in Table 2 the statement %HTML{...%} encompasses the HTML statements which are generated by the dynamic HTTP application. In the example shown in Table 2, the HTML generated is based on

substitution of the contents of four variables for those variables. See %HTML(purpose){...%} for example. The first variable is "header" which provides header information for a header to be included in the web page to be generated. The second variable is "contents." Here, the value of the variable "contents," which is discussed above, is inserted in the web page. Accordingly the variable "contents" in this example provides the Table of Contents to be displayed on the left-hand column of the web page, as shown in Fig. 4. In a frame-based system, which would require a frame-capable browser, the Table of Contents would correspond to a separate frame for displaying table of contents information. However, as seen here, separate frames are not required since the information can be displayed in a single web page generated by the dynamic HTTP application 32. The third variable in the HTML section is the variable "purpose". Here, the value of the variable "purpose" is inserted into the web page by the dynamic HTTP application, when the macro is interpreted. As shown in Fig. 4, the paragraphs defined as the values of the variable "purpose" are included in the HTML for the web page sent to the client 36. Accordingly, the web browser 38 displays those paragraphs in an area that would correspond to a main frame in a frame-based system. The last variable in the HTML section for this example is the variable "footer". Here, a footer is produced by generating corresponding HTML tags for the web page by substituting the contents of the variable "footer", resulting in a

HTML sections (step S120). In this case, the first HTML section corresponds to "purpose," and accordingly that portion of the HTML part of the web macro is interpreted by the dynamic HTTP application 32. When the first HTML section is interpreted, the value of the variable "purpose" is substituted for the variable "purpose", along with values of other specified variables being substituted (step S130). The result is the web page shown in Fig. 4 being generated (step S140). The web page shown in Fig. 4 includes the header 58, the table of contents 50, and the footer 59, and it also includes in a middle portion, or main area 56, the purpose information that is defined for the "purpose" variable. Here, the value of the "purpose" variable is shown in the middle portion of the display and includes the word "Purpose" and two paragraphs of text. In this manner when a requestor selects an item from the first area 50, corresponding to the Table of Contents in this example, a request is sent to the web server 30 and ultimately to the dynamic HTTP application 32 to generate another web page which includes not only the information of the first area 50 (e.g., the table of contents information) but also second information corresponding to information displayed in the main area 56, here, the Purpose information. Once the HTML web page is generated and transmitted from the web server 30 to the client 36 (step S150), the web browser 38 displays the HTML page as shown in Fig. 4 (step S160). Other information such as header 58 and footer 59 information can also be included in the generated web page, for presentation at the client by the web browser.

same header 58, footer 59 and table of contents 50 presented in the previous page, but now also presenting new information shown in the main area, namely, the Overview information 66.

In the manner described above, information is defined only once, in the variable definitions, and can be maintained in a single file, yet multiple web pages can be generated from that single instance of the information. Accordingly, the information which is used in multiple web pages can be maintained in only a single location, thereby eliminating the need to maintain a separate copy of that information for each web page in which it is used. Maintaining all the information for the web site in a single computer file simplifies the webmaster's task of having to keep track of all the separate files which comprise a web page, or web site. Moreover, by only providing those portions of the web page requested by a user, maintaining all the information for the web page or web site does not impact performance, since only the requested portions are transmitted. Furthermore, the single instance of the information stored in the web macro can be handled and presented with a non-frame capable web browser. Such a non-frame capable web browser can present the information, maintained in a single instance, because the client receives a complete web page in response to each selection of an item in the first area, rather than receiving only a frame of information. Accordingly, a web page generated according to the present invention and presented at the client using a non-frame capable web browser, gives the appearance of consisting of logical

server 30 which pass it to the dynamic HTTP application. The dynamic HTTP application executes the HTML section and evaluates the "if" statement. Accordingly, the value of the variable "overview" is substituted for the \$(overview) statement in the HTML section of the web macro, since the argument passed to the HTML section is "overview." In this manner the same HTML section can be used for all the items listed in the Table of Contents area of the web page.

Table 3: Web Macro with Single HTML Section

```
%define{
  header = {
    <center>IBM Internet Yellow Pages</center>
  }
  contents = {
    <b>Table of Contents</b>
    <ul>
      <li><a href="...">Purpose</a>
      <li><a href="...">Overview</a>
      <li>Hot <a href="...">New Technology</a>
      .
      .
      .
    </ul>
  }
  purpose = {
    <b><i>Purpose</i></b>
    <blockquote>
      To rapidly communicate material and ...
      <p>This internet site is updated ...
    </blockquote>
  }
  overview = {
    <b><i>Overview</i></b>
    <blockquote>
      IBM's Internet Yellow Page Solution is a collection of IBM
      software and hardware products....
      <p>Using IBM's award-winning database, DB2, ...
    </blockquote>
  }
  footer = {
    <a>IBM home page</a> |
    <a>Order</a> |
    .
    .
    .
  }
}</BODY></HTML>
```


Table 3 (continued)

●

% }

```
%HTML_INPUT{
```

\$ (header)

\$(contents)

```
%if ( "$arg" == "purpose" )
```

\$(purpose)

```
%elseif ( "$arg" == "overview" )
```

\$ (overview)

•

●

\$(footer)

% }

In still another embodiment the web browser 38 is controlled by the web macro and web server to make the presentation of the information in the web macro 34 appear even more like frames, by including borders around the various logical frames, or presentation areas, of the web page, as shown in Fig. 7. More specifically, the HTML generated based on the web macro 34 causes the web browser to create a pseudo-border 82 around each of the presentation areas defined in the HTML section, as shown in Fig. 7. Here, a border is drawn around one or more of the presentation areas of the web page. The border is defined by the HTML generated to produce the web page. The web page is constructed by the dynamic HTTP application 32 so that the client continues to receive a full web page upon each request. However, in this embodiment borders are also produced to more clearly delineate the boundaries between the different presentation areas displayed at the client.

In yet another embodiment, the web macro 34 allows a web page to be generated which include a plurality of presentation areas and a background. Here, the background is decoupled from the plurality of presentation areas so that the background appears uniform across the presentation areas. For example, if the web macro includes a background tag for pages that are generated, upon receiving a request for the Purpose information discussed above, the dynamic HTTP application would generate a web page having header, footer, Table of Contents and Purpose presentation areas, along with a single uniform background defined in

1 the web macro. Fig. 8 shows an example of a web page created in which the
2 macro, in the HTML section for the requested information, defines a background
3 80 to be displayed. As can be seen in Fig. 8, the background 80 appears uniform
4 across all the presentation areas. This capability would not be possible were the
5 web page implemented with frame tags.

6 In still yet another embodiment, the variables are defined in one or more
7 different files. By use of "include" statements, the values of the variables can be
8 retrieved from the files in which they are stored and included in the single web
9 macro containing the HTML sections when that macro is invoked.

10 Other modifications and variations to the invention will be apparent to
11 those skilled in the art from the foregoing disclosure and teachings. Thus, while
12 only certain embodiments of the invention have been specifically described herein,
13 it will be apparent that numerous modifications may be made thereto without
14 departing from the spirit and scope of the invention. For example, the present
15 invention is described in terms of generating HTML as a language for producing
16 the web page. However, the present invention is not limited to only the HTML
17 language. Rather, the present invention includes any other language capable of
18 being dynamically generated to express materials to be presented at a client. Also,
19 the present invention is not limited to an HTTP application, but includes any other
20 dynamic application capable of dynamically substituting values of variables to
21 create presentation materials.